

A Cascaded Feature Pyramid Network with Non-Backward Propagation for Facial Expression Recognition¹

Wei Yang, Hongwei Gao, Yueqiu Jiang, Jiahui Yu, Jian Sun, Jinguo Liu, *Senior Member, IEEE*, and Zhaojie Ju, *Senior Member, IEEE*

Abstract—In this work we propose a novel cascaded feature pyramid network with non-backward propagation (CFPN-NBP) for facial expression recognition (FER) that addresses the problems inherent in traditional backward propagation (BP) algorithms in the training process by using the Hilbert-Schmidt independence criterion (HSIC) bottleneck. The proposed algorithm is developed at two different levels. At the first level, a novel training method HSIC bottleneck is considered as an alternative to traditional BP optimization, where the correlation between the output of the hidden layers and the input, and the correlation between the output of the hidden layers and its label are calculated to reduce redundant information; hence, the least information is used to predict the results. At the second level, a novel architecture is designed in the feature extraction process. The convolutional layers with the same resolutions are densely connected and introduced into the attention mechanism, so that the model can focus on more important information. The convolutional layers with different resolutions are combined by three cascaded pyramid networks; in this way, the shallow features and the deep features can be further fused, and; therefore, the semantic information and the content information can both be reserved. To further reduce the number of parameters, the operation of separable convolution instead of traditional convolution is utilized. Experiments on the challenging FER2013 dataset show that the proposed CFPN-NBP algorithm improves the accuracy of the FER task and outperforms the related state-of-the-art methods.

Index Terms—Cascaded feature pyramid network, facial expression recognition, HSIC bottleneck, non-backward propagation, separable convolution

I. INTRODUCTION

FACIAL expression recognition also known as classification, is the task of annotating images with semantic labels. It is one of the most active fields in robotics. Current state-of-the-art FER algorithms commonly use deep convolutional neural networks (DCNNs) with backward propagation optimization [1]. Deep learning has brought a new level of performance to an increasingly wide range of image classification tasks, but the

process of backward error propagation is currently considered biologically unreasonable [2]–[4] and does not conform to the signal propagation process of the neural networks of the human brain. Specifically, the related gradient descent algorithms (e.g. stochastic gradient descent (SGD)) are time-consuming and occupy a large amount of memory, due to the need to constantly search for suitable super parameters (e.g., learning rate). Therefore, an improved HSIC-bottleneck method to theoretically and practically replace the gradient backward propagation algorithm of neural networks is proposed, and to make better use of the extracted features, a novel architecture named the cascaded feature pyramid network (CFPN) is designed.

In this paper, the expression of the loss function is determined by maximizing the mutual information between the output of the hidden layers and the label, and minimizing the interdependence between the output of hidden layers and the input data, so that the least input information would be used to predict the output; at the same time, redundant features are removed. The performance on large and small objects is more remarkable. In this paper, the FER2013 dataset is used for our experiments. Compared with the traditional gradient backward propagation algorithm, the convergence rate is significantly improved, while the accuracy is boosted, and the generalization is enhanced, moreover, the computation burden and memory usage are greatly reduced.

The main contributions are summarized as follows:

- 1) A novel method is proposed to effectively replace backward propagation optimization. The new method is more reasonable with the biological theory. In particular, the redundant information can be removed, promoting the high efficiency of training; therefore, the convergence rate is further improved.
- 2) A novel architecture named CFPN is proposed. It is a classic yet powerful architecture that makes full use of extracted features by cascaded pyramids. The attention mechanism is further improved by assigning a corresponding weight to each feature in the dense block and a feature fusion operation is used to represent the importance of the feature. The feature maps with the same

This work was supported in part by Liaoning Province Higher Education Innovative Talents Program Support Project (LR2019058), Liaoning Revitalization Talents Program (XLYC1902095), Joint Funds of the National Natural Science Foundation of China (No. 51575412, U1609218), and CAS Interdisciplinary Innovation Team (Grant No. JCTD-2018-11).

Corresponding authors: Hongwei Gao, Zhaojie Ju.

Wei Yang, Hongwei Gao, Yueqiu Jiang and Jian Sun are with the School of Automation and Electrical Engineering, Shenyang Ligong

University, Shenyang, China (e-mail: xcfyangwei@163.com, ghw1978@sohu.com, yueqiujiang@sylu.edu.cn, jiaoyou2017sj@163.com).

Jinguo Liu is with the Shenyang Institution of Automation, Chinese Academy of Sciences, Shenyang, China, (e-mail: liujinguo@sia.cn)

Jiahui Yu and Zhaojie Ju are with the School of Computing, University of Portsmouth, PO13HE, UK (e-mail: jiahui.yu@port.ac.uk, zhaojie.ju@port.ac.uk).

scale are connected by dense blocks, and the feature maps with different scales are connected by cascaded pyramids. The high-level features, which are semantically strong but of lower resolution, are upsampled and combined with higher resolution features to generate feature representations that are both of high resolution and semantically strong.

- 3) Separable convolution is used instead of traditional convolution operations to extract not only the spatial information but also the channel feature. At the same time, the number of parameters is significantly reduced, so that the algorithm is easier to implement on a mobile device.

The remainder of the paper is organized as follows: Section II briefly reviews related work, focusing on the traditional FER algorithms, the typical architecture of CNN models and the optimization method. Section III describes the proposed CFPN-NBP in detail. Section IV reports our experiments and results analysis, and finally, Section V summarizes our main conclusions and highlights future work.

II. RELATED WORK

A. The Facial Expression Recognition Task

As a signal for human beings to express emotion directly, facial expression is one of the most natural pieces of information. The task of FER has been analyzed through a series of studies due to its significance in the field of human-computer interaction. According to Y. -I. Tian et. al. [5], in the daily communication of human beings, the information transmitted through language and sound accounts for 7% and 38% of the total information, respectively. Hand gestures also play an important role in human interactions. In 2018, B Liao et al. [6] proposed a hand gesture recognition method and used the RGB-D dataset collected by Intel RealSense Front-Facing Camera SR300. The amount of information transmitted through facial expressions accounts for 55%. American psychologists Ekman and Friesen defined six typical expressions through a series of related studies [7]: happiness, anger, surprise, fear, disgust, and sadness.

The FER task can be mainly divided into two categories depending on the type of input: static image FER and dynamic sequence FER. For the *static recognition task* [8]-[10], the semantic information is encoded according to a single image input, while for the *dynamic recognition task* [11], [12], the representation of hidden layers is related to the temporal relation among contiguous frames in the input facial expression sequence. In 2019, J Li et al. [13] used Microsoft Kinect to collect the RGB-D dataset and used a two-stream network to implement a *dynamic recognition task*. In this paper, we will limit our discussion on FER based on the *static recognition task*.

B. Traditional Facial Expression Recognition Algorithms

Traditional facial expression recognition algorithms are mainly based on manually extracted features. In 2013, Shaohua Wan et al. [14] used the Gabor transformation to extract features, which reduced the impact of facial pose changes on recognition accuracy. In 2015, Ali et al. [15] used the empirical mode decomposition (EMD) technique to perform facial expression recognition, projected two-dimensional images continuously to obtain facial feature maps, and used EMD

techniques to decompose facial feature maps. In addition, Shin et al. [16] performed expression recognition on the emotional dimension using independent component analysis (ICA) and principal component analysis (PCA). Lajevardi et al. [17] developed a three-dimensional color frame into a two-dimensional matrix, extracted features using a Lob-Gabor filter, and classified the features using a linear discriminant analysis classifier. Feature extraction in the process of traditional facial expression recognition relies heavily on human intervention, and the algorithm has poor robustness and low precision.

C. Convolutional Neural Network for Classification

At this stage, deep learning, as a novel field of machine learning research, has received widespread attention. Deep learning has significantly improved the timeliness and accuracy. CNN is a typical algorithm for deep learning. In 1989, LeCun et al. [18] proposed the idea of the earliest CNN algorithm, and in 1998, they proposed an algorithm using CNN to implement handwritten digit recognition. In 2014, Karen Simonyan et al. [19] used the VGG network to win second place in the ILSVRC competition. The entire network used a small convolutional kernel (3x3) instead of a large convolutional kernel. In the same year, Christian Szegedy et al. [20] used the Inception V1 model to win first place in the ILSVRC competition. This is an important milestone in the history of CNN network development. The model uses the inception module, and the architectural decision is based on the Hebb principle [21] and multiscale processing. The above purpose is to design a network with excellent local topology, that is, to perform multiple convolutional operations or pooling operations on the input image in parallel. The internal computing resources of the network are better utilized, and the burden of calculations is greatly reduced, but because the model is a deep network, it will encounter problems i.e. vanishing gradient and exploding gradient. In 2015, Christian Szegedy et al. [22] proposed Inception V2, which introduced batch normalization (BN) in the network, so that the output of each layer is standardized as a Gaussian distribution. The accuracy of the model is further improved and drawn on the method of VGG in the use of multiple consecutive small convolution kernels instead of single large convolution kernels, which reduces the number of parameters, and strengthens the representation of hidden features of the network. In the same year, Christian Szegedy et al. [23] proposed the Inception V3 network, which further improved on the basis of the original network. First, the two-dimensional convolutional kernel was split into two one-dimensional convolutional kernels, which on the one hand improves the nonlinearity of hidden features and enhances the expression ability of the model; and on the other hand, reduces the training parameters and therefore avoids the overfitting problem. Split asymmetric convolutional kernel structures can handle more abundant spatial features and increase the diversity of features. Second, the inception module structure was optimized in Inception V3, and the design of 35x35, 17x17 and 8x8 modules were more elaborate, further enhancing its adaptability of the scale, but the problems in Inception V2 were not further solved. In December 2015, Kaiming He et al. [24] proposed the residual neural network (ResNet), which won the championship in ILSVRC 2015 competition. The residual module is introduced in the network, that is, by directly passing

the input information to the output without any operations, an identity mapping is added, so that the gradient of the next layer of the network can be directly transmitted to the upper layer in the process of backward propagation, which solves the problem of the vanishing gradient of the deep neural network, and makes the network go deeper. In 2016, Christian Szegedy et al. [25] proposed Inception V4, which is based on Inception V3; it kept the basic structure of the network unchanged, but made the network wider and deeper so that more features could be learn, and the performance was further improved. In the same year, Christian Szegedy et al. [25] also designed Inception ResNet, mainly using the shortcuts method of the ResNet to accelerate the training of the deep neural networks, and the performance of the network was also further improved. In 2017, Huang Gao et al. [26] proposed DenseNet. The main idea of DenseNet proposed in the paper was to establish the connection relationship between different layers so that the extracted features were fully utilized, and the problem of vanishing gradient was further avoided. DenseNet also made the network narrower, the number of parameters compared to other models was significantly reduced, and the training efficiency is improved at the same time. In 2020, J. Yu et al. [27] used improved Inception ResNet layers for automatic recognition, which further improved the performance of the DCNN algorithm, but it could not merge the multilevel and multiscale features. Thus, we proposed the CFPN architecture to further improve the performance of the FER algorithm.

III. PROPOSED APPROACH

At present, the major optimization method of convolutional neural networks at home and abroad is the stochastic gradient descent algorithm, which uses the direction of the negative gradient obtained from the global classification error to decompose the global optimization task into several subproblems to update the weights and biases layer by layer. Therefore, it has many shortcomings, i.e., large training parameters, a high order of computational power needed and a huge complexity of computation in the very deep model structure.

The rationality of gradient backward propagation algorithms in biology has already been a controversial topic and a motivation for exploring other alternatives. An obvious nonconforming problem in the gradient backward propagation algorithms is that the synaptic weights are adjusted according to the error of the latter layer, which is unreasonable in biological theory [28], [29]. Another problem is that in forward propagation and backward propagation, the weight matrix is shared [30], [31], and the backward propagation is linearly calculated and must be stopped when calculating forward propagation [32]. Therefore, finding a more reasonable alternative for backward propagation has become an important issue in the field of deep learning (DL), and it is also an urgent requirement.

In 2019, Wan-Duo Kurt Ma et al. [33] proposed a method to replace backward propagation, citing HSIC metrics, and using sampling to measure the strength of two distribution dependencies. This method avoids the vanishing gradient and exploding gradient phenomena. Cross-layer optimization is possible without gradient backward propagation in any layer. It

can simultaneously optimize multiple layers in parallel; however, the effectiveness of small object detection in complex convolutional neural networks still has a gap in traditional backward propagation training algorithms. Therefore, an improved HSIC-bottleneck method to replace traditional backward propagation training is proposed in this paper. Based on the Hilbert space kernel method, the raw data are mapped to the kernel function in reproducing kernel Hilbert space (RKHS), and then the covariance operator is constructed to describe the conditional independence. According to the theorem of conditional independence, the objective function of measure independence and conditional independence can be obtained. In the actual operation, the empirical condition covariance operator is constructed according to the sample data, and the estimation function represented by the Gram matrix can be obtained by the inner product operation in RKHS.

A. Problem Formulation and Method Overview

Information theory is the basis of learning theory and extensive research [34]. The information bottleneck (IB) principle [35] generalizes the notion of the minimal sufficient statistics, expressing a tradeoff in the hidden representation between the information needed for predicting the output and the information retained about the input. The optimal solution can be given by:

$$\min_{p_{O_i|X, PY|O_i}} I(X; O_i) - \beta I(O_i; Y) \quad (1)$$

where X, Y represent the input and label respectively, and O_i represents the output features of layer i . It should be noted that the information characterized by O_i relating to Y is extracted from X . β represents the Lagrange multiplier, $I(X; O_i)$ is the mutual information between X and O_i , and $I(O_i; Y)$ is the mutual information between O_i and Y . As seen from the formula, IB mainly retains the output information about the label in the hidden layer when compressing the feature information of the input data.

In practice, IB is difficult to calculate for many reasons. If the input signal is regarded as continuous, i.e., a voice signal, unless a noise signal is added to the network, the mutual information $I(X; O_i)$ is infinite, so many algorithms divide the input data into bins so that the data will not be extended to high dimensions, but this would give rise to different results due to different binning sizes. Additional influencing factors are the difference between discrete and continuous data, and between discrete data and differential entropy. The HSIC is used instead of mutual information terms in of the IB objective in this paper. Unlike mutual information estimation, HSIC adopts time complexity $O(l^2)$ with a robust computing method, in which l represents the amount of input data.

HSIC is the Hilbert-Schmidt norm of the cross-covariance operator between the data distributions in the RKHS [36] formed as:

$$\begin{aligned} HSIC(P_{MN}, H, G) &= \|C_{mn}\|^2 \\ &= E_{MNM'N'}[k_m(m, m')k_y(y, y')] \\ &\quad + E_{MM'}[k_m(m, m')]E_{NN'}[k_n(n, n')] \\ &\quad - 2E_{MN}\{E_{M'}[k_x(m, m')]E_{N'}[k_n(n, n')]\} \end{aligned} \quad (2)$$

where k_m, k_n represents the kernel function, H, G represents the Hilbert space, and E_{MN} represents the expectation of MN . Referring to (2), the following expression can be derived:

$$HSIC(P_{mn}, H, G) = (l - 1)^{-1} \text{tr}(K_M H K_N H) \quad (3)$$

where l represents the number of samples, $K_M \in R^{l \times l}$, $K_N \in R^{l \times l}$, $K_{M_{ij}} = k(m_i, m_j)$, $K_{N_{ij}} = k(n_i, n_j)$, $H \in R^{l \times l}$ is a centrally symmetric idempotent matrix, and $H = I_l - \frac{1}{l} \begin{pmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{pmatrix}_{l \times l}$. In this way, the calculation cost is only related to the number of samples, which has an advantage over calculating high-dimensional small sample data.

In a fully connected network with h hidden layers, the output matrix dimension of the hidden layer is $(1, d_i)$, where $i \in \{1, \dots, h\}$, d_i is the number of hidden units in the hidden layer i , and the hidden layer output matrix size of each batch is (b, d_i) , where b is the batch size. In the process of applying the IB principle to calculate the objective function, HSIC can be used instead of mutual information as:

$$Z_i^* = \underset{Z_i}{\text{argmin}} HSIC(Z_i, X) - \beta HSIC(Z_i, Y) \quad (4)$$

where Z_i represents the hidden representation, Z_i^* represents the optimal hidden representation, X is the input data, Y is the label, and β is the Lagrange multiplier. According to (3), items of HSIC in (4) can be expressed as:

$$HSIC(Z_i, X) = (l - 1)^{-1} \text{tr}(K_{Z_i} H K_X H) \quad (5)$$

$$HSIC(Z_i, Y) = (l - 1)^{-1} \text{tr}(K_{Z_i} H K_Y H) \quad (6)$$

Referring to (4), (5) and (6), the optimal hidden representation Z_i^* finds a balance between redundant information that does not depend on the input and the maximum correlation with the output. In ideal conditions, when (4) converges, the information needed to predict the label is preserved and redundant information leading to overfitting is removed.

B. Cascaded Feature Pyramid Network with Attention Mechanism

An important observation in previous works in object detection is that it is necessary to “merge” features at different scales [37]. The cross-scale connections allow the model to merge high-level features with strong semantics and low-level features with high resolution. There are five convolutional layers in each block, and different weight vectors are assigned to the output feature map of each layer during concatenating. In the process of forward propagation, each layer is connected with all other layers in the dense block, that is, each layer connects the output feature maps of all previous layers as its own input, and their own output is passed to all subsequent layers, enhancing the flow of feature maps among the layers, making full use of the extracted feature maps. Compared with DenseNet, the dense block in this paper introduces the connection weight. In this way, the model can pay more attention to important features. The output feature of layer i can be calculated as:

$$o_i = T_i W_i([o_0, o_1, \dots, o_{i-1}]) \quad (7)$$

where T_i is the composite function of layer i , including BN, ReLU and separable convolution. W_i is the weight vector of layer i . o_i is the output feature of layer i . The dense block structure is shown in Fig. 1. There are two steps in the process

of the separable convolution operation, as shown in Fig. 2, and takes the number of input feature channels is five. Step one uses the same number of kernels as the channels of the input feature in the convolution operation. Each kernel only has a single channel. Then the input feature is convoluted channel-by-channel, and the number of output feature maps is the same as the channels of the input feature. Step two uses the kernel with a size of 1×1 and the same channel number as the input feature in the convolution operation. Then, step one’s output in the dimension of depth is collected. For example, take the number of input feature channels as five, the filter size as 3×3 and the number of filters as three. If the common convolution operation is used, the number of parameters is calculated as:

$$P_{\text{num}} = K_w \times K_H \times K_c \times K_{\text{num}} \quad (8)$$

where P_{num} is the number of trainable parameters, K_w and K_H is the kernel size, K_c is the channel number of kernels, and K_{num} represents the number of kernels.

Use of the separable convolution operation can significantly reduce the number of parameters. In step one, the number of trainable parameters is calculated as:

$$P_{\text{num}_1} = K_{w_1} \times K_{H_1} \times K_{c_1} \times K_{\text{num}_1} \quad (9)$$

In step two, the trainable parameters are calculated as:

$$P_{\text{num}_2} = K_{w_2} \times K_{H_2} \times K_{c_2} \times K_{\text{num}_2} \quad (10)$$

Then the total number of trainable parameters can be calculated as:

$$P_{\text{total}} = P_{\text{num}_1} + P_{\text{num}_2} \quad (11)$$

Using separable convolution can further improve the convergence rate of the model. To make full use of the extracted feature maps, inspired by M2Det [38], we propose a novel architecture by using dense CNN, FPN and an attention mechanism. The architecture of the proposed model is shown in Fig. 3. In Fig. 3 (i), we illustrate the structure of the whole model. The size of the input data has been resized to $224 \times 224 \times 1$ from $48 \times 48 \times 1$, and then the input data are passed to a dense CNN module to extract features. The architecture of the dense CNN is shown in Fig. 3 (ii), which has 4 dense blocks and the scale of the features is reduced by half compared to the previous block. The scale of the output feature can be calculated as:

$$\begin{aligned} O_H &= \frac{I_H + 2 \times P_H - F_H}{S_H} \\ O_W &= \frac{I_W + 2 \times P_W - F_W}{S_W} \end{aligned} \quad (12)$$

where O_H and O_W are the height and width of the output, respectively; I_H and I_W are the height and width of the input, respectively; P_H and P_W are the height and width of the padding, respectively; F_H and F_W are the height and width of the filter, respectively; and S_H and S_W are the height and width of the filter stride, respectively.

The output feature size of the first dense block is $112 \times 112 \times 128$, the output feature size of the second dense block is $56 \times 56 \times 128$, the output feature size of the third dense block is $28 \times 28 \times 128$, and the output feature size of the last dense block is $14 \times 14 \times 128$. We fuse the output of the last block and the penultimate block to generate the base feature, as shown in Fig. 3 (iii). The output of the last dense block becomes $14 \times 14 \times 64$ by the separable convolution operation, then upsampled to $28 \times 28 \times 64$, and the output of the penultimate dense block

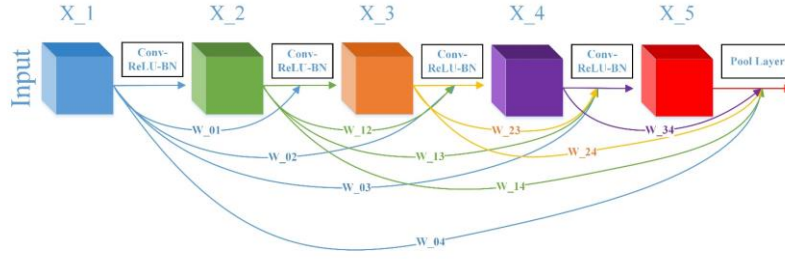


Fig. 1. The structure of each dense block with five convolutional layers, where each convolutional layer indicates separable convolution operation, ReLU is the activation function, BN is batch normalization, W_{ab} is the weight matrix of layer “a” when concatenating at the layer “b”, and X_i is the input feature of layer i .

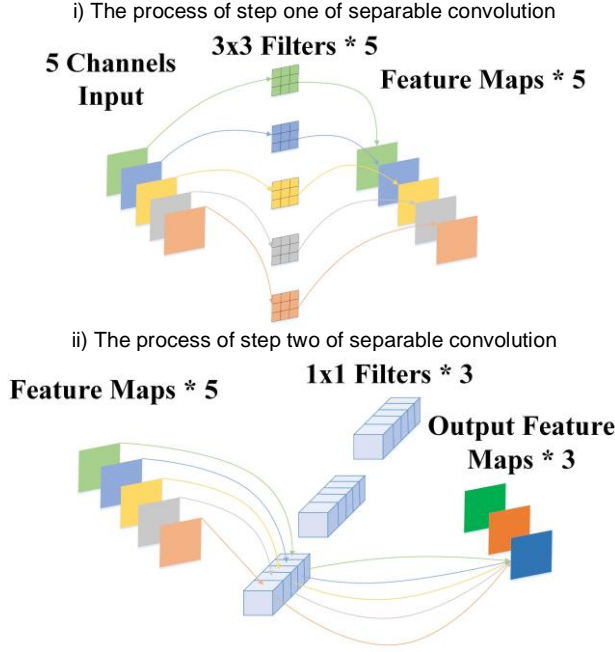


Fig. 2. Subfigure (i) presents the process of step one of the separable convolution operation. Subfigure (ii) presents the process of step two of the separable convolution operation. Both take the number of the input feature channels as five.

becomes $28 \times 28 \times 64$ by the separable convolution operation, so that both features have the same scale, then they can concatenate at the channel dimension to generate the base feature. The size of the base feature is $28 \times 28 \times 128$. The operation of “Fusion_1” can be explained as:

$$Fusion_1 = Cat\{Us[T_1(x_1)], T_2(x_2)\} \quad (13)$$

where Cat is the concatenation operation. Up is the upsample operation. T_i is the composite function, including BN, ReLU and separable convolution. x_1 is the output of the last dense block. x_2 is the output of the penultimate dense block.

Then, the base feature is passed to the cascaded FPN to further extract features, as shown in Fig. 3 (v). There are three pyramid networks in the cascaded FPN module to generate multiscale and multilevel feature maps, and there are five levels of feature maps in each pyramid. We use the separable convolution operation with stride 2 to reduce the size of feature maps and the number of parameters. The high-level features with strong semantics but lower resolution are concatenated with the higher resolution features to generate feature representations that are both high resolution and semantically strong. The max feature map upsampled with a size of $28 \times 28 \times 64$ in the “Pyramid_1” network is fused with the

base feature by the “Fusion_2” operation as the input feature of “Pyramid_2”, as shown in Fig. 3 (iv). The operation of “Fusion_2” can be represented as:

$$Fusion_2 = Cat[T(Base\ Feature), O_i] \quad (14)$$

where Cat is the concatenation operation. T is the composite function, including BN, ReLU and separable convolution. O_i is the max output feature map of upsampled pyramid i .

The channel of the base feature is reduced to 64, and the max feature map of upsampled “Pyramid_1” has the same size as the base feature, so that they can be concatenated to generate a feature with a size of $28 \times 28 \times 128$ as the input of “Pyramid_2”. The max output feature map of upsampled “Pyramid_2” has the same size as the base feature, and hence, they can be concatenated by the “Fusion_2” operation and used as the input of “Pyramid_3”. The output feature of “Pyramid_1” is regarded as the shallow feature, the output feature of “Pyramid_2” is regarded as the medium feature and the output feature of “Pyramid_3” is regarded as the deep feature. The output features of FPN are passed to the “Fusion_3” operation, and the attention mechanism is introduced to generate features with a size of $7 \times 7 \times 192$ by a global attention operation on the combined features, as shown in Fig. 3 (vi). The “Fusion_3” operation can be represented as:

$$\begin{aligned} X &= Cat(O_1, O_2, O_3) \\ W &= F_2\{F_1[GAP(X)]\} \\ Fusion_3 &= XW \end{aligned} \quad (15)$$

where Cat is the concatenation operation. GAP is the global average pooling operation. F_i is the composite function, including BN, ReLU and full connection. W is the attention vector that can promote the model to focus on more important channels and removes redundant information.

The output feature maps of FPN with a size of $7 \times 7 \times 64$ are concatenated to generate the feature with a size of $7 \times 7 \times 192$, which contains multilevel features from different depths. However, the performance of the simple concatenation operation is not sufficient, so we add a channel-oriented attention mechanism by using global average pooling (GAP). The feature map then has a size of $1 \times 1 \times 192$, the statistics of channels with global features are obtained and then passed to the full connection (FC) layers on the channel dimension to obtain the different attentions for different channels. This attention mechanism encourages feature maps to focus on the channels that they benefit the most. The hidden units of the first FC layer are one-eighth of the channel of the input feature, and the hidden units of the second FC layer are eight times the number of units of the first FC layer, so that the output of the

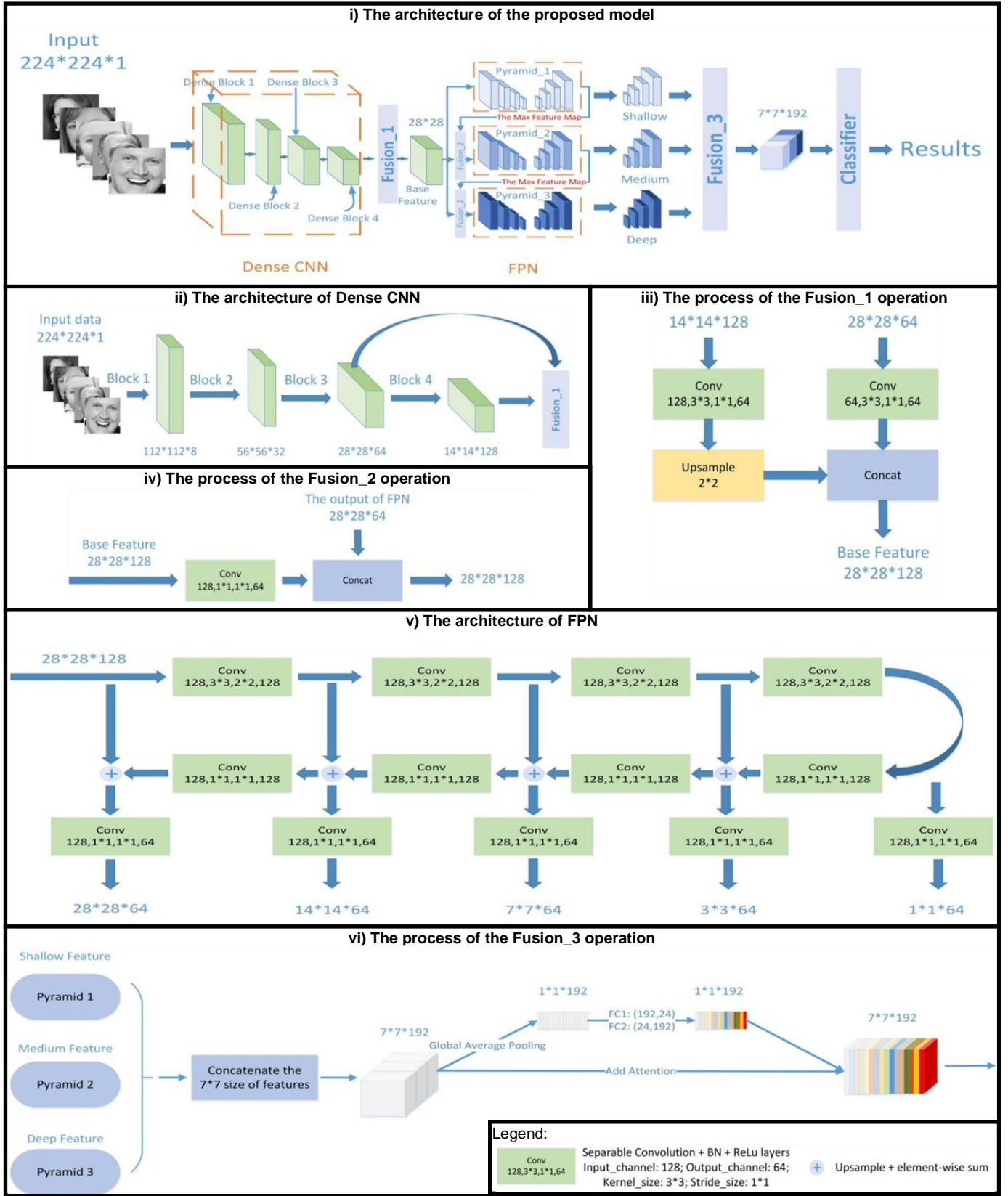


Fig. 3. Subfigure (i) presents the architecture of the whole proposed model which mainly consists of five parts. The input data size is $224 \times 224 \times 1$. Subfigure (ii) presents the architecture of Dense CNN which has four blocks and the output feature maps are fused by the "Fusion_1" operation. Subfigure (iii) presents the details of the process of the "Fusion_1" operation, the output of the last dense block and the output of the penultimate dense block are combined to generate the base feature. Subfigure (iv) presents the process of the "Fusion_2" operation which combines the base feature and the output of the pyramid network. Subfigure (v) presents the architecture of each pyramid in FPN, in each pyramid, there are five levels of feature maps. Sub-figure (vi) presents the process of the "Fusion_3" operation, where the attention is added on the combined feature.

second FC layer can be resized to match the size of the input of the FC layers. Then, the first step of the separable convolution operation is used to add attention to the combined features channel-by-channel.

IV. EXPERIMENTAL STUDY

A. Datasets and Experimental Setup

Our experiments were performed on the FER2013 public dataset for the 2013 Kaggle competition. The dataset is stored in the form of a csv file. Compared with other static FER datasets, the most important difference is that the FER2013 dataset has already aligned the faces, and it contains expressions for almost every situation and include men, women, children and cartoon characters. Therefore, we chose the most representative dataset to implement our experiments. The data information includes the labels, pixel values, and usage (training, validation and test) of the facial expression recognition task. A sample of the dataset is shown in Fig. 4. There are seven different labels that present seven emotions as shown in Table. I.

emotion		pixels	Usage
0	0	70 80 82 72 58 58 60 63 54 58 60 48 89 115 121...	Training
1	0	151 150 147 155 148 133 111 140 170 174 182 15...	Training
2	2	231 212 156 164 174 138 161 173 182 200 106 38...	Training
3	4	24 32 36 30 32 23 19 20 30 41 21 22 32 34 21 1...	Training
4	6	4 0 0 0 0 0 0 0 0 0 0 3 15 23 28 48 50 58 84...	Training

Fig. 4. The sample of the dataset, including the serial number of data, labels, pixel values and usage.

TABLE I
THE LABELS AND THE CORRESPONDING EXPRESSIONS

Label	Emotion
0	Anger
1	Disgust
2	Fear
3	Happy
4	Sad
5	Surprised
6	Normal

According to the usage, the dataset is divided into three parts, namely, the training set, validation set and test set, including 28708 data points in the training set, 3589 data points in the validation set and the same number of data points in the test set. The training set, validation set and test set samples are shown in Fig. 5.

i) A sample of the training dataset			ii) A sample of the development dataset		
	pixels	emotion		pixels	emotion
0	70 80 82 72 58 58 60 63 54 58 60 48 89 115 121...	0	0	170 118 101 88 88 75 78 82 66 74 68 59 63 64 6...	0
1	151 150 147 155 148 133 111 140 170 174 182 15...	0	1	7 5 8 6 7 3 2 6 5 4 4 5 7 5 5 5 6 7 7 7 10 10 ...	5
2	231 212 156 164 174 138 161 173 182 200 106 38...	2	2	232 240 241 239 237 235 246 117 24 24 22 13 12...	6
3	24 32 36 30 32 23 19 20 30 41 21 22 32 34 21 1...	4	3	200 197 149 139 156 89 111 58 62 95 113 117 11...	4
4	4 0 0 0 0 0 0 0 0 0 0 3 15 23 28 48 50 58 84...	6	4	40 28 33 56 45 33 31 78 152 194 200 186 196 20...	2

iii) A sample of the test dataset		
	pixels	emotion
0	254 254 254 254 254 254 255 160 2 58 53 70 77 ...	0
1	156 184 198 202 204 207 210 212 213 214 215 21...	1
2	69 118 61 60 96 121 103 87 103 88 70 90 115 12...	4
3	205 203 236 157 83 158 120 116 94 86 155 180 2...	6
4	87 79 74 66 74 96 77 80 80 84 83 89 102 91 84 ...	3

Fig. 5. Subfigure (i) presents a sample of the training set, including the serial number of the data, pixel values and the labels. Subfigure (ii) presents a sample of the development set. Subfigure (iii) presents a sample of the test set.

The distribution histograms of the labels of the training set, validation set, and test set are shown in Fig. 6.

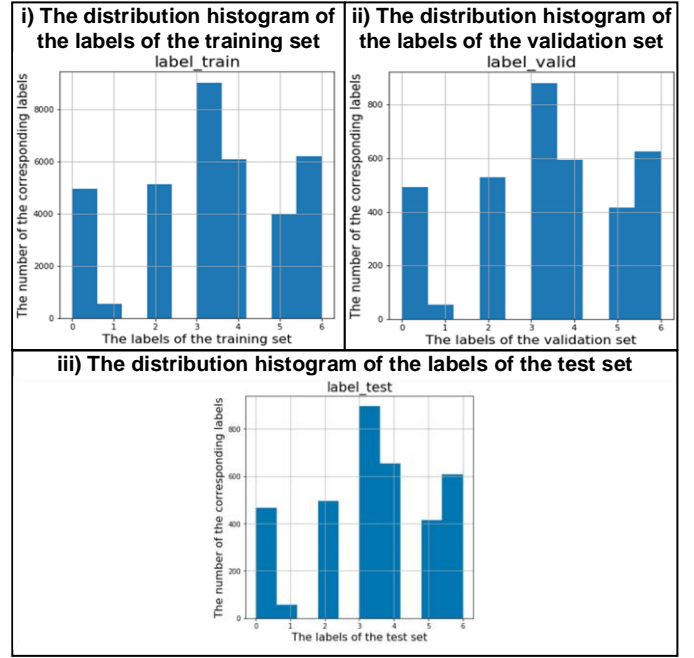


Fig. 6. Subfigure (i) presents the distribution histogram of the labels of the training set and corresponds to the distribution of the number of the seven kinds of expressions. Subfigure (ii) presents the distribution histogram of the labels of the valid set. Subfigure (iii) presents the distribution histogram of the labels of the test set.

The pixel values in the data are restored to an image with a size of $48 \times 48 \times 1$; a single channel grayscale of the facial expression dataset include seven expressions: anger, disgust, fear, happy, sad, surprised and normal. The corresponding image samples are shown in Fig. 7.



Anger Disgust Fear Happy Sad Surprised Normal

Fig. 7. The samples of 7 kinds of facial expressions for the FER2013 dataset.

We use the training set to train the various methods of our study and improve them on the validation set. Finally, we evaluated them on the test set. To evaluate the performance of the proposed algorithm on the FER2013 dataset, we use the accuracy of the test set and compare our results with those of Alexnet, ResNet-50, Inception Net V2 and DenseNet.

The experiment in this paper is based on the Python language. We use Python version 3.6, mainly with the TensorFlow and Keras frameworks. The version of TensorFlow is tensorflow-gpu 1.11.0. Tensorboard and Matplotlib are used to visualize

the results. The hardware CPU platform uses an Intel Core I7-9700k, the GPU uses a single NVIDIA GeForce RTX- 2080, and the memory is 8 GB.

B. Implementation Details

The input data are resized to $224 \times 224 \times 1$ by a bilinear interpolation operation from $48 \times 48 \times 1$. The location of the face, the light, the angle of the shot, and other factors can all affect the classification results, so to improve the accuracy and generalization performance of the proposed algorithm, we use data augmentation operations on the training dataset to increase the amount of dataset by using horizontal flipping, 45 degrees clockwise rotation, 45 degrees counterclockwise rotation, half scaling, double scaling, and adding Gaussian noise operations. The expanded dataset is seven times larger than the original.

In this paper, the performance of traditional backward propagation training networks and non-backward propagation networks using the same parameters are compared. The convergence rate and the final classification results under different learning rates are tested. Finally, the results of the proposed algorithm are compared with the state-of-the-art algorithms using the same hyperparameters, and the performance of the algorithm is verified. We set the Lagrange multiplier of the HSIC-bottleneck β to 100 based on experience during training.

C. Preliminary Experiments - Design Choices

Through a large number of experiments, we find the best way of using the proposed algorithm with the non-backward propagation method by comparing different parameters and intermediate versions of them, as shown in Fig. 3 (i).

To verify the non-backward propagation algorithm proposed in this paper, the accuracy and loss value of the proposed method and the traditional backward propagation training are shown in Fig. 8.

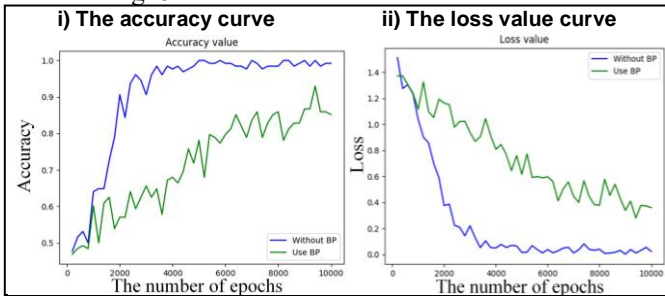


Fig. 8. Subfigure (i) presents the accuracy curve of the non-backward propagation network and BP network in the process of training. Subfigure (ii) presents the loss value curve of the non-backward propagation network and BP network in the process of training.

In Fig. 8 (i) and Fig. 8 (ii), we use a batch size of 128 and a learning rate of 0.005. The final accuracy rate reaches 0.9946. It can be seen from Fig. 8 that the traditional BP algorithm has not reached convergence at 10,000 epochs, while the non-backward propagation algorithm achieves convergence at 2500 steps, and the non-backward propagation network has higher accuracy. The proposed HSIC-bottleneck can train each layer separately, allowing each layer to be optimized separately, without passing forward gradients, achieving parallel computing. The results when using different activation function experiments are shown in Fig. 9.

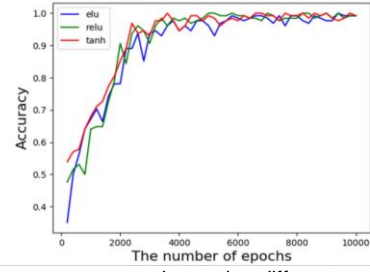


Fig. 9. The accuracy curve when using different activation functions

As shown in Fig. 9, the results obtained using the tanh function and the elu function are almost the same but not as good as those obtained using the ReLU function.

The performance of the proposed algorithm in the training process under different learning rates is shown in Fig. 10. As seen from Fig. 10, the convergence rate of the model is almost the same when the learning rate is 0.001 and 0.005. However, when the learning rate is set to 0.005, the model performs best. When the learning rate is set to 0.0005, the convergence rate of the model is significantly slower. Therefore, to achieve better accuracy and latency tradeoff, the optimal learning rate is determined through a series of experiments.

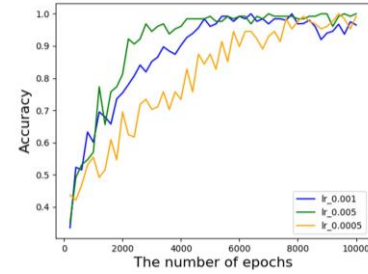


Fig. 10. The performance of the proposed algorithm under different learning rates.

In this paper, the proposed algorithm is compared with Alexnet, Inception v2, Resnet 50 and DenseNet. The learning rate is set at 0.005, and the batch size is 128. The results under the same environment and parameters are shown in Fig. 11.

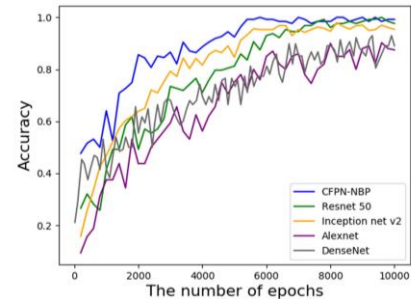


Fig. 11. The accuracy curve of different models under the same environment and parameters.

As seen in Fig. 11, it is obvious that the convergence rate of the proposed algorithm is faster than that of the other algorithms. Compared with backward propagation, the HSIC-bottleneck separates the hidden signals in the representation of individual neurons, which indicates that the object of the HSIC-bottleneck helps to make the distribution of extracted features more independent and easier to associate with their labels. The per-step running time and the final results on the test set of the proposed algorithm and other state-of-the-art algorithms are shown in Table II.

TABLE II
THE RESULTS AND PER STEP RUNNING TIME OF DIFFERENT METHODS

Method	Per step running time	The accuracy in test set
Alexnet ^[39]	0.243178s	0.74
Inception v2 ^[22]	0.565114s	0.81
Resnet-50 ^[24]	0.180212s	0.85
DenseNet ^[26]	0.324537s	0.83
CFPN-NBP	0.169286s	0.94

As seen from Table II, the algorithm proposed in this paper not only achieves the accuracy of the traditional backward propagation algorithm, but also greatly improves the training speed. In addition, the algorithm proposed in this paper requires less computation, and occupies less memory, which proves that the algorithm has better performance and that the optimization method is feasible.

V. CONCLUSION AND FEATURE WORK

In this study, a CFPN-NBP algorithm is proposed, in which the HSIC-bottleneck is used instead of traditional gradient backward propagation. The proposed algorithm further improves the performance of CNN and does not require manual intervention, using the Adam optimizer but without backward propagation. The results of the experiments show that compared with other traditional deep learning algorithms, the CFPN-NBP algorithm has great advantages in training speed, accuracy and calculation in facial expression recognition tasks. However, end-to-end learning tasks often require a large dataset, which poses a great challenge to the field of expression recognition lacking label data; additional depth information could potentially improve the performance. These aspects will be the main focus in the future research [40], [41].

REFERENCES

- [1] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550-1560, Oct. 1990.
- [2] J. M. Kleinberg, "Two algorithms for nearest-neighbor search in high dimensions," in *Proc. of 29th Annual ACM Symposium on Theory of Computing (STOC 1997)*, EL Paso, Texas, USA, 1997, pp. 599-608.
- [3] B. Widrow, A. Greenblatt, Y. Kim, and D. Park, "The No-Prop algorithm: A new learning algorithm for multilayer neural networks," *Neural Networks*, vol. 37, pp. 182-188, Jan. 2013.
- [4] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" *Advances in Neural Information Processing Systems (NIPS 2014)*, 2014, pp. 3320-3328.
- [5] Y.-I. Tian, T. Kanade, and J. F. Cohn, "Recognizing action units for facial expression analysis," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 23, no. 2, pp. 97-115, Feb. 2001.
- [6] B. Liao, J. Li, Z. Ju, and G. Ouyang, "Hand gesture recognition with generalized hough transform and DC-CNN using realsense," in *8th International Conference on Information Science and Technology (ICIST 2018)*, Cordoba, Spain, 2018, pp. 84-90.
- [7] P. Eckman, "Universal and cultural differences in facial expression of emotion," *Nebraska symposium on motivation*, vol. 19, pp. 207-284, Jan. 1972.
- [8] P. Liu, S. Han, Z. Meng, and Y. Tong, "Facial expression recognition via a boosted deep belief network," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Sept. 2014, pp. 1805-1812.
- [9] A. Mollahosseini, D. Chan, and M. H. Mahoor, "Going deeper in facial expression recognition using deep neural networks," in *2016 IEEE Winter Conf. on Applications of Computer Vision (WACV)*, Lake Placid, NY, USA, 2016, pp. 1-10.
- [10] Z. Ju, X. Ji, J. Li, and H. Liu, "An Integrative Framework of Human Hand Gesture Segmentation for Human-Robot Interaction," *IEEE Systems Journal*, vol. 11, no. 3, pp. 1326-1336, Sept. 2017.
- [11] G. Zhao, and M. Pietikainen, "Dynamic texture recognition using local binary patterns with an application to facial expressions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 915-928, April 2007.
- [12] B. Liu, Z. Ju, and H. Liu, "A structured multi-feature representation for recognizing human action and interaction," *Neurocomputing*, vol. 318, pp. 287-296, Nov. 2018.
- [13] J. Li, Y. Mi, and G. Li, "CNN-Based Facial Expression Recognition from Annotated RGB-D Images for Human-Robot Interaction," *International Journal of Humanoid Robotics*, vol. 16, no. 4, pp. 17, Aug. 2019.
- [14] S. Wan, and J. K. Aggarwal, "A scalable metric learning-based voting method for expression recognition," in *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, Shanghai, China, 2013, pp. 1-8.
- [15] H. Ali, M. Hariharan, S. Yaacob, and A. H. Adom, "Facial emotion recognition using empirical mode decomposition," *Expert Systems with Applications*, vol. 42, no. 3, pp. 1261-1277, Feb. 2015.
- [16] Y.-s. Shin, "Recognizing facial expressions with pca and ica onto dimension of the emotion," in *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, Hong Kong, China, 2006, pp. 916-922.
- [17] S. M. Lajevardi, and Z. M. Hussain, "Emotion recognition from color facial images based on multilinear image analysis and Log-Gabor filters," in *2010 25th International Conference of Image and Vision Computing New Zealand, Queenstown, New Zealand*, 2010, pp. 1-6.
- [18] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, Nov. 1998.
- [19] K. Simonyan, and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014. [Online]. Available: <https://arxiv.org/abs/1409.1556>
- [20] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov et al., "Going deeper with convolutions," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, 2015, pp. 1-9.
- [21] M. D. Plumbley, "Efficient information transfer and anti-Hebbian neural networks," *Neural Networks*, vol. 6, no. 6, pp. 823-833, 1993.
- [22] S. Ioffe, and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, 2015. [Online]. Available: <http://arxiv.org/abs/1502.03167>
- [23] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 2818-2826.
- [24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 770-778.
- [25] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proc. of the 31st AAAI Conference on Artificial Intelligence*, San Francisco, California, USA, Feb. 4-10, 2017.
- [26] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017, pp. 4700-4708.
- [27] J. Yu, H. Gao, W. Yang, Y. jiang, W. Chin, N. Kubota et al., "A Discriminative Deep Model with Feature Fusion and Temporal Attention for Human Action Recognition," *IEEE Access*, vol. 8, pp. 43243-43255, March 2020.
- [28] W. Xiao, H. Chen, Q. Liao, and T. Poggio, "Biologically-plausible learning algorithms can scale to large datasets," *CoRR*, vol. abs/1811.03567, 2018. [Online]. Available: <https://arxiv.org/abs/1811.03567>
- [29] T. P. Lillicrap, D. Cownden, D. B. Tweed, and C. J. Akerman, "Random synaptic feedback weights support error backpropagation for deep learning," *Nature Communications*, vol. 7, no. 1, pp. 1-10, Nov. 2016.
- [30] S. Grossberg, "Competitive learning: From interactive activation to adaptive resonance," *Cognitive Science*, vol. 11, no. 1, pp. 23-63, March 1987.

- [31] T. P. Lillicrap, D. Cownden, D. B. Tweed, and C. J. Akerman, "Random feedback weights support learning in deep neural networks," *CoRR*, vol. abs/1411.0247, 2014. [Online]. Available: <https://arxiv.org/abs/1411.0247>
- [32] Y. Bengio, D.-H. Lee, J. Bornschein, T. Mesnard, and Z. Lin, "Towards biologically plausible deep learning," *CoRR*, vol. abs/1502.04156, 2015. [Online]. Available: <https://arxiv.org/abs/1502.04156>
- [33] W.-D. K. Ma, J. P. Lewis, and W. B. Kleijn, "The HSIC Bottleneck: Deep Learning without Back-Propagation," *CoRR*, vol. abs/1908.01580, 2019. [Online]. Available: <http://arxiv.org/abs/1908.01580>
- [34] T. M. Cover, *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. New York, NY, USA: Wiley, 2006, pp. 14-25.
- [35] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," *arXiv preprint physics/0004057*, 2000.
- [36] A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf, "Measuring statistical dependence with Hilbert-Schmidt norms," in *International Conference on Algorithmic Learning Theory*, Singapore, 2005, pp. 63-77.
- [37] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017, pp. 2117-2125.
- [38] Q. Zhao, T. Sheng, Y. Wang, Z. Tang, Y. Chen, L. Cai *et al.*, "M2det: A single-shot object detector based on multi-level feature pyramid network," in *Proc. of the 31st AAAI Conference on Artificial Intelligence*, Honolulu, Hawaii, USA, Jan. 27-Feb. 1, 2019.
- [39] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. of the 25th International Conference on Neural Information Processing Systems (NIPS 2012)*, Lake Tahoe, NV, USA, Dec. 2012, pp. 1097-1105.
- [40] B. Liu, H. Cai, Z. Ju, H. Liu, RGB-D sensing based human action and interaction analysis: a survey, *Pattern Recognition*. Vol. 94, pp. 1-12, 2019
- [41] J. Yu, H. Gao, W. Yang, W. Chin, N. Kubota, Z. Ju, A discriminative deep model with feature fusion and temporal attention for human action recognition, *IEEE Access*, Vol. 8, p. 43243-43255, 2020.



Wei Yang was born in Heilongjiang, China, in 1994. He received his B.S. degree in automation from Shenyang Ligong University of China in 2016 and his M.S. in automation and electrical engineering: detection technology and automatic equipment from Shenyang Ligong University. His research interests include deep learning, pattern recognition and digital image processing.



intelligent computation. He has published more than sixty technical papers in these areas as first authors or co-authors.

Hongwei Gao received his Ph.D. degree in the field of pattern recognition and intelligent system from Shenyang Institute of Automation (SIA), Chinese Academy of Sciences (CAS) in 2007. Since September 2015, he has been a professor of School of Automation and Electrical Engineering, Shenyang Ligong University. Currently, he is the leader of academic direction for optical and electrical measuring technology and system. His research interests include digital image processing and analysis, stereo vision and

intelligent computation. He has published more than sixty technical papers in these areas as first authors or co-authors.

Yueqiu Jiang received the B.S. and M.S. degrees in computer science from Shenyang Ligong University, China, in 1998 and 2001, respectively, and the Ph.D. degree in computer science from Northeast University, China, in 2004. She worked as a lecturer from March 2004 to Aug. 2006 in School of Science, Shenyang Ligong University, and worked as an associate professor from Sep. 2006 to Aug. 2010 in School of Science, Shenyang Ligong University. Currently, she is a professor in School of

Information Science and Engineering, Shenyang Ligong University. Her research interests include image processing, multimedia applications and satellite communications and signal processing.



Jiahui Yu received his B.S. and M.S. degrees in intelligent systems from Shenyang Ligong University, China, in 2017 and 2019, respectively. Since 2019, he is currently working toward the Ph.D. degree at University of Portsmouth, U.K. His current research interests include machine intelligence, pattern recognition, and human-robot/computer interaction and collaboration.



Jian Sun received his B.S. and M.S. degrees in control engineering from Shenyang Ligong University, China, in 2016 and 2019, respectively. Since 2019, he is currently working toward the Ph.D. degree at Shenyang Ligong University, China. His current research interests include machine learning, digital image processing, pattern recognition, and signal processing.



Jinguo Liu (M'07-SM'18) received his Ph.D. degrees in Mechatronics from Shenyang Institute of Automation (SIA), Chinese Academy of Sciences (CAS) in 2007. Since January 2011, he has been a Full Professor with SIA, CAS. He is also holding the Assistant Director position of State Key Laboratory of Robotics from 2008 and the Associate Director position of Center for Space Automation Technologies and Systems from 2015. His research interests include bio-inspired robotics and space robot.

He has authored/coauthored three books, over one hundred papers and fifty patents in above areas. He was awarded the T. J. TARN Best Paper Award in Robotics from 2005 IEEE International Conference on Robotics and Biomimetics, the Best Paper Award of the Chinese Mechanical Engineering Society in 2007, the Best Paper Nomination Award from 2008 International Symposium on Intelligent Unmanned Systems, the Best Paper Award from 2016 China Manned Space Academic Conference, and the Outstanding Paper Award from 2017 International Conference on Intelligent Robotics and Applications, and the Best Paper Award from 2018 International Conference on Electrical Machines and systems. He is a senior member of the IEEE, IEEE Technical Committee on Safety, Security, and Rescue Robotics, IEEE Technical Committee on Marine Robotics, and the Senior Member of Chinese Mechanical Engineering Society.



Zhaojie Ju (M'08-SM'16) received the B.S. in automatic control and the M.S. in intelligent robotics both from Huazhong University of Science and Technology, China, and the Ph.D. degree in intelligent robotics at the University of Portsmouth, UK. He held a research appointment at the University College London, London, U.K., before he started his independent academic position at the University of Portsmouth, U.K., in 2012. His research interests include machine intelligence, pattern recognition, and their

applications on human motion analysis, multi-fingered robotic hand control, human-robot interaction and collaboration, and robot skill learning. He has authored or co-authored over 180 publications in journals, book chapters, and conference proceedings and received four best paper awards and one Best AE Award in ICRA2018. Dr. Ju is an Associate Editor of several journals, such as IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON COGNITIVE AND DEVELOPMENTAL SYSTEMS and Neurocomputing.